CLAIMS

1. A method comprising:

calling a remote object that resides on a server using an object-oriented network protocol;

receiving a reply from the server, the reply containing server state information; and

caching the server state information for use in subsequent communication with the server.

2. A method as recited in claim 1, wherein the server state information is embodied as a data object.

3. A method as recited in claim 1, wherein the server state information comprises a service ID and data.

4. A method as recited in claim 1, wherein the object-oriented network protocol comprises a remote procedure call (RPC) protocol.

5. A method as recited in claim 1, further comprising:

subsequently calling the remote object using the object-oriented network protocol; and

submitting the server state information to the remote object.

6. A method comprising:

receiving a request for a local program object from a remote application program interface on a requesting computer;

creating a state-caching object that contains state information pertaining to the request;

processing the request to generate a reply; and

returning the reply together with the state-caching object to the requesting computer.

7. A method as recited in claim 6, wherein the state-caching object contains a service ID field to identify a computer or group of computers that created the state-caching object and a data field.

8. A method as recited in claim 6, further comprising inserting the state-caching object into the process state used to process the request.

9. A method as recited in claim 6, further comprising:

subsequently receiving another request from the requesting computer, along with the state-caching object; and

using the state-caching object to recall state information pertaining to a previous connection with the requesting computer.

10. A method comprising:

submitting a request to a server using a non-HTTP protocol over a network;

receiving a reply from the server, the reply containing a state-caching object with session state information; and

storing the state-caching object for use in subsequent communication with the server.

11.   A method as recited in claim 10, wherein the state-caching object comprises a service ID and data.

12.   A method as recited in claim 10, wherein the non-HTTP protocol comprises an object-oriented network protocol.

13.   A method as recited in claim 10, wherein the non-HTTP protocol comprises a remote procedure call (RPC) protocol.

14.   A method as recited in claim 10, further comprising:

submitting a subsequent request to the server; and

sending the state-caching object along with the subsequent request.

15.   A method comprising:

routing a request from a first computer to a second computer via a network;

routing a reply from the second computer back to the first computer via the network, the reply carrying state information of the second computer that pertains to the request; and

maintaining the state information within the network.

16. A method as recited in claim 15, wherein the server state information is embodied as a data object.

17. A method as recited in claim 15, wherein the network comprises one or more network components and the maintaining comprises storing the state information of the second computer on one of the network components.

18. A method as recited in claim 15, wherein the network comprises multiple network components and the maintaining comprises continually passing the state information among the network components.

19. A method as recited in claim 15, wherein the network supports remote procedure call protocol and the routing a request comprises passing the request as part of a call to a program object located on the second computer.

20. A method comprising:

performing request/reply exchanges among multiple computers organized in a computer cluster;

generating state-caching objects that contain state information of corresponding computers as part of the request/reply exchanges;

storing the state-caching objects on one or more different computers within the computer cluster to maintain the state information remotely from the corresponding computers from which the state-caching objects originated and preserve the state information in an event that one of the corresponding computers fails.

21. A method as recited in claim 20, wherein the performing request/reply exchanges comprises conducting remote procedure calls to remote program objects on another computer.

22. A method as recited in claim 20, further comprising, in an event that one of the corresponding computers fails, using the state-caching object associated with the failed computer to at least partially restore state information for the failed computer during recovery.

23. A method as recited in claim 20, wherein each state-caching object contains a service ID field to identify a service as represented by one or more computers that created the state-caching object and a data field.

24. A stateless distributed computer architecture, comprising:

a program object resident at a first computing device;

an application program interface (API) resident at a second computing device to facilitate calls to the program object at the first computing device using an object-oriented network protocol;

the program object, responsive to a call, returning a reply with a state-caching object that contains state information pertaining to the first computing device; and

wherein the state-caching object is stored on the second computing device for later communication with the first computing device.

25. A stateless distributed computer architecture as recited in claim 24, wherein the object-oriented network protocol comprises a remote procedure call (RPC) protocol.

26. A stateless distributed computer architecture as recited in claim 24, wherein the first and second computer are organized in a cluster of computers.

27. A method as recited in claim 24, wherein the state-caching object contains a service ID field to identify the service of the first computing device and a data field.

28. A stateless distributed computer system, comprising:

a network having one or more network components to route requests from a first endpoint device to a second endpoint device and to route replies from the second endpoint device back to the first endpoint device, wherein at least one reply contains state information pertaining to the second endpoint device; and

the network being configured to maintain the state information and to reassociate the state information with a subsequent request from the first endpoint device to the second endpoint device.

29. A stateless distributed computer system as recited in claim 28, wherein at least one of the network components stores the state information.

30. A stateless distributed computer system as recited in claim 28, wherein multiple network components continually route the state information amongst themselves to preserve the state information.

31. A computer-readable medium comprising computer-executable instructions that, when executed on one or more processors, direct a computing device to:

call a remote object that resides on a remote computer using an object-oriented network protocol;

receive a reply from the remote computer, the reply containing state information of the remote computer; and

cache the state information for use in subsequent communication with the remote computer.

32. A computer-readable medium as recited in claim 31, wherein the state information is embodied as a data object.

33. A computer-readable medium as recited in claim 31, wherein the state information comprises an identity of the remote computer, a network endpoint identity, an identity of the remote object, and data.

34. A computer-readable medium as recited in claim 31, wherein the object-oriented network protocol comprises a remote procedure call (RPC) protocol.

35. A computer-readable medium as recited in claim 31, further comprising computer-executable instructions that, when executed on one or more processors, direct a computing device to:

subsequently call the remote object using the object-oriented network protocol; and

submit the state information to the remote object.

36. A computer-readable medium comprising computer-executable instructions that, when executed on one or more processors, direct a computing device to:

create a state-caching object that contains state information pertaining to a request for a local program object received from a remote application program interface;

generate a reply; and

return the reply together with the state-caching object to the client.

37. A computer-readable medium as recited in claim 36, wherein the state-caching object contains a computer ID field to identify a computer that created the state-caching object, an endpoint ID field to identify a network endpoint, an object ID field to identify the local program object, and a data field.

38. A computer-readable medium as recited in claim 36, further comprising computer-executable instructions that, when executed on one or more processors, direct a computing device to insert the state-caching object into a context thread used to process the request.

39. A computer-readable medium as recited in claim 36, further comprising computer-executable instructions that, when executed on one or more processors, direct a computing device to:

subsequently receive the state-caching object in a subsequent request; and

use the state-caching object to recall the state information.

40. A computer-readable medium comprising computer-executable instructions that, when executed on one or more processors, direct a computing device to:

submit a request to a remote computer using a non-HTTP protocol over a network;

receive a reply from the remote computer, the reply containing session state information; and

cache the session state information for use in subsequent communication with the remote computer.

41. A computer-readable medium as recited in claim 40, wherein the session state information comprises a remote computer ID, a network endpoint ID, and data.

42. A computer-readable medium as recited in claim 40, wherein the non-HTTP protocol comprises an object-oriented network protocol.

43. A computer-readable medium as recited in claim 40, wherein the non-HTTP protocol comprises a remote procedure call (RPC) protocol.

44. A computer-readable medium as recited in claim 40, further comprising computer-executable instructions that, when executed on one or more processors, direct a computing device to:

submit a subsequent request to the remote computer; and

send the session state information along with the subsequent request.

45. A computing device comprising:

means for calling a remote object that resides on a remote computer using an object-oriented network protocol;

means for receiving a reply from the remote computer, the reply containing state information pertaining to the remote computer; and

means for caching the state information for use in subsequent communication with the remote computer.

46. A computing device comprising:

means for receiving a request for a local program object from a remote application program interface;

means for creating a state-caching object that contains state information pertaining to the request;

means for generating a reply; and

means for returning the reply together with the state-caching object to the client.

47.   A network comprising:

means for routing a request from a first computer to a second computer;

means for routing a reply from the second computer back to the first computer, the reply carrying state information of the second computer that pertains to the request; and

means for maintaining the state information on behalf of the first and second computers.